N94-14640

Space Station Common Berthing Mechanism, a Multi-Body Simulation Application

Ian Searle
Boeing Defense and Space Group

July 30, 1992

1 Introduction

This paper discusses an application of multi-body dynamic analysis conducted at the Boeing Company in connection with the Space Station (SS) Common Berthing Mechanism (CBM). After introducing the hardware and analytical objectives we will focus on some of the day-to-day computational issues associated with this type of analysis.

1.1 Hardware

The major components of the CBM are the four 5-bar mechanisms, two berthing port contact rings, and the alignment guides. The function of the CBM is to complete the attachment of two modules once the modules are in close proximity. The modules are initially placed with either the Space-Station Remote Manipulator System (SSRMS) or the Shuttle Remote Manipulator System (SRMS). Once the modules are in position the RMS is put in *limp-mode*, which signifies that all RMS actuators are disabled and the arm can be driven by external forces to a new position. Moving the RMS while it is in the limp-mode means that some of the RMS actuators must be back-driven. The amount of force required to back-drive an RMS is a function of the arm position, which determines what combination of RMS joints must be driven.

After the RMS has been placed in limp-mode, and any residual motion has ceased[1], the final berthing sequence is initiated. The four capture latch mechanisms extend, then retract in an effort to grab the passive berthing

[1] New requirements specify berthing operations shall tolerate small residual motions.

module's latches. As the two modules are being pulled together, the alignment guides force the two modules into precise alignment through contact. Once the modules are in contact, a set of powered bolts[2] fastens the two modules together.

Figure 1 shows the active and passive ports of the CBM[3]. The active port contains eight alignment guides and the four capture-latch mechanisms. The passive port contains four alignment guides, which fit tightly between the active port guides when the two ports are berthed. Typical module and station weights can range from 30,000 lbf to 450,000 lbf.

Figure 2 depicts the capture-latch mechanism. The mechanism consists of four links: the two drive-arms, the idler-arm, and the capture-arm. Only one of the drive-arms is actually driven; we shall call this the primary drive-arm. The other pivots freely about the drive shaft; we shall call this the secondary drive-arm. The primary drive-arm is powered by a DC motor with a purely mechanical clutch, which limits the applied torque. The motor control attempts to drive the motor at a constant speed of 2 rpm. At the time the analysis was performed the motor-clutch design was not final. It was assumed that the rate sensor for the motor would be placed on the output shaft of the clutch. The capture-latch mechanism has a total weight of approximately 5 lbf.

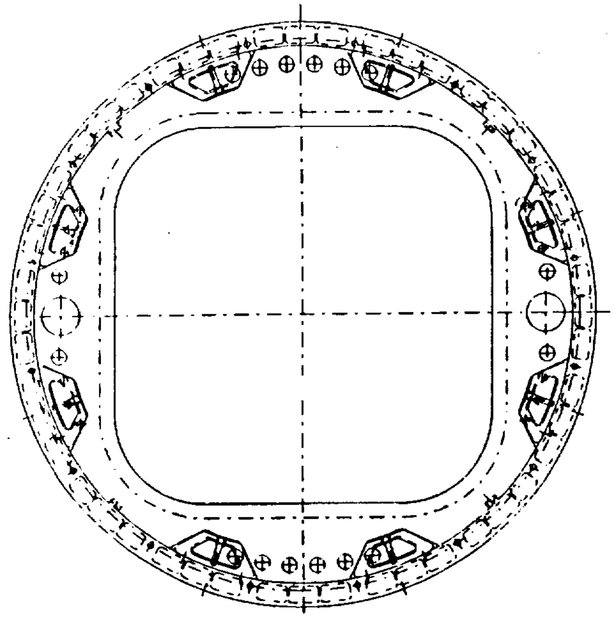
Figure 3 depicts the capture-latch mechanism motion for a typical berthing. Initially the mechanism is in the open state. When the mechanism is commanded to close, the capture-arm swings into the passive port and travels down the edge until it engages the passive port capture-latch fitting. The mechanism drive arm continues to rotate at 2 rpm until the drive-arm is over-center and the ports are in contact.

2 Objective

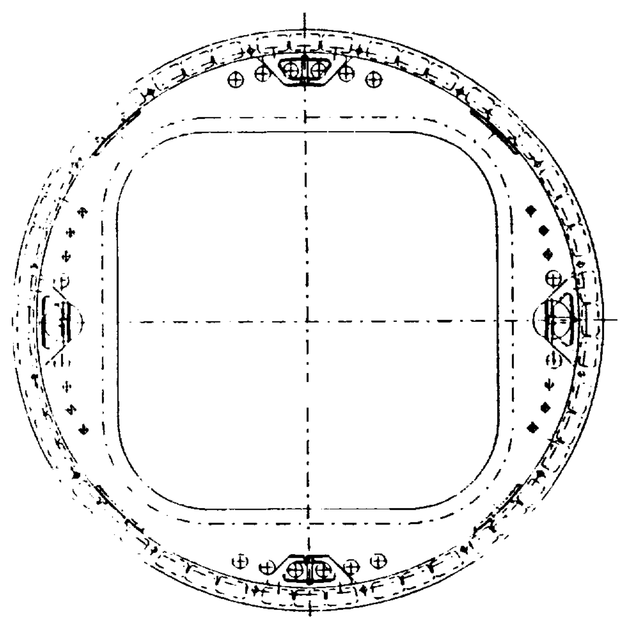
The principle objective of the analysis effort was to assess the capability of the CBM to function in the presence of SSRMS back-drive forces, friction, and worst-case port-to-port misalignment. The capture-latch mechanism drive motor and clutch have a slip-torque-limit of 40 in-lbf. The definition of worst-case SSRMS back-drive forces had not been finalized. In order to

[2]For the rest of this paper will will ignore the powered bolts since they are not a part of this analysis.

[3]The CBM alignment guide configuration was not finalized at the time the analysis was performed



Active Alignment Guides
PDR-I configuration.



Passive Alignment Guides
PDR-I configuration.

Figure 1: Common Berthing Mechanism, Active and Passive Ports

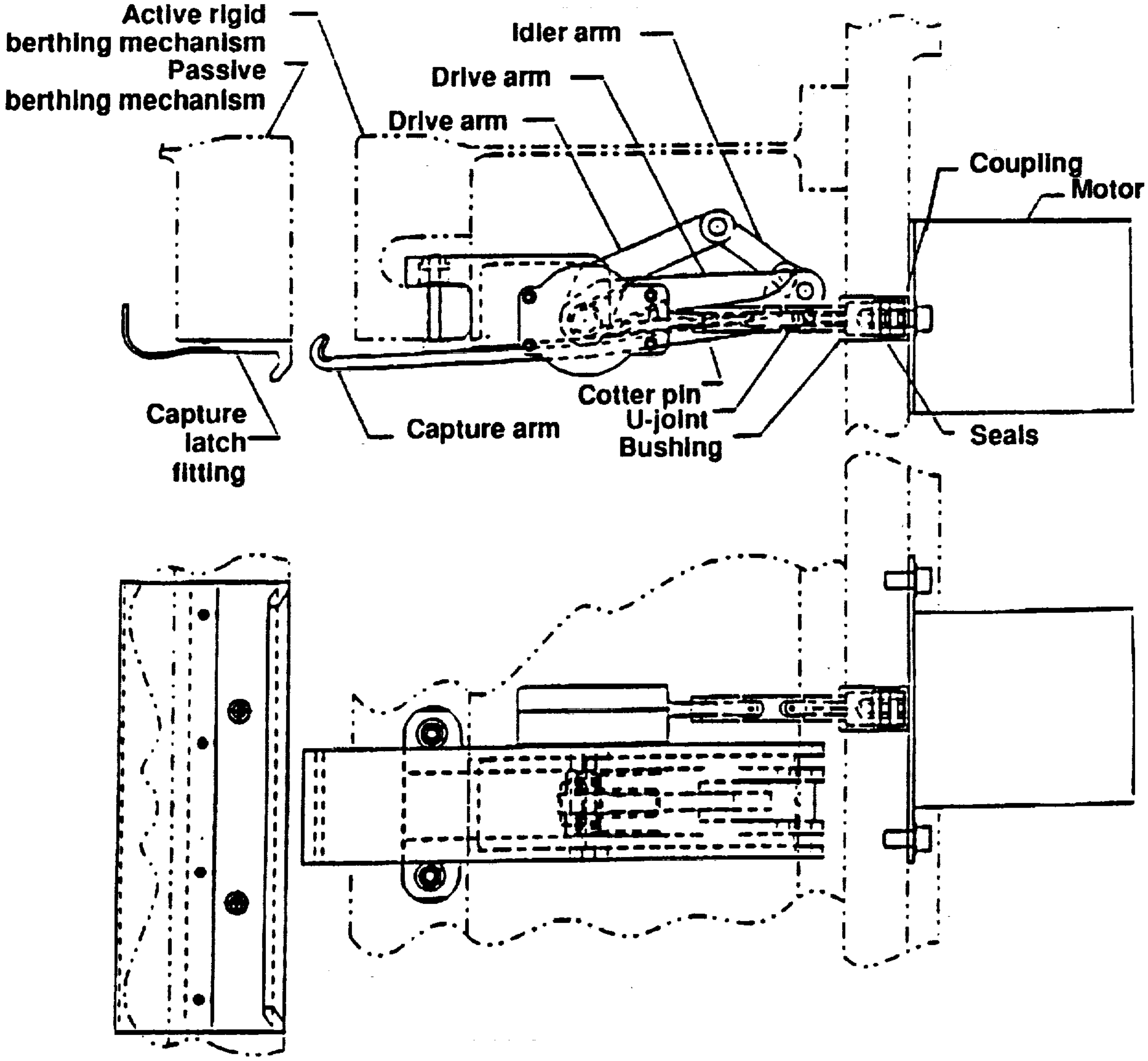


Figure 2: Common Berthing Mechanism, Capture-Latch Mechanism

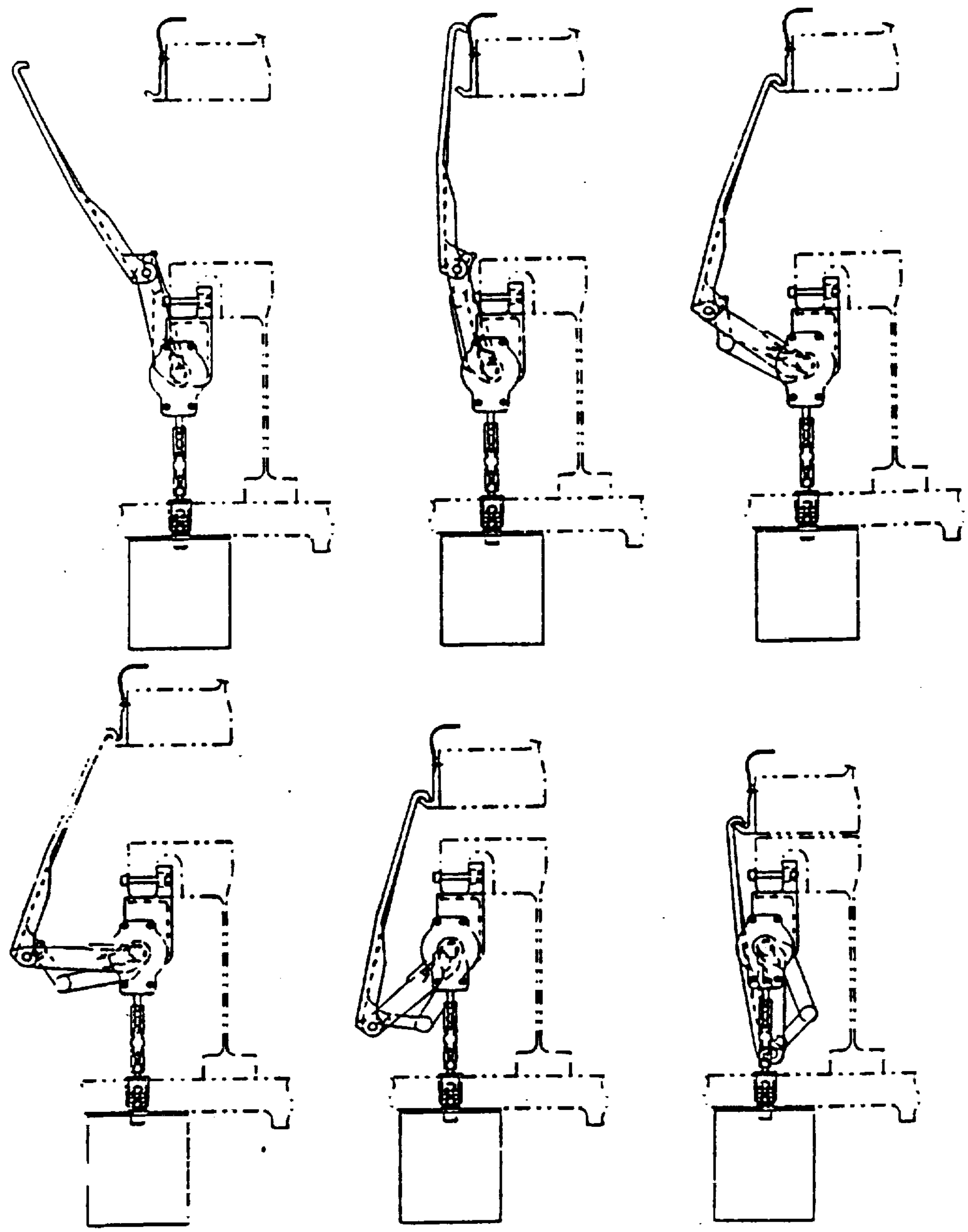


Figure 3: Common Berthing Mechanism, Capture-Latch Mechanism Motion

finalize the design, an analytical capability to assess the design in the face of changing requirements was necessary.

3 Decisions

At this point we had to decide how best to construct a simulation capability in a short period of time. Listed below are most of the options that were considered for this task, and a few words explaining each option's strong and/or weak points.

1. A Boeing in-house FORTRAN code exists that is well developed for transient dynamics problems, but has very little constraint capability. All degrees of freedom (d.o.f.) must be expressed in a common global coordinate system. This means that a mechanism, whether it is in a plane of the global coordinate system or not, must have more degrees of freedom than are required. For this particular problem we have four 5-bar mechanisms, thus we will have to have 96 d.o.f. just for the mechanisms. Furthermore, we will have to add "soft-constraints" to remove the d.o.f that we did not want in the first place. This program is very good at some problems, but this does not appear to be one of them; both the size of the problem, and the 80 plus extra opportunities to make a mistake were significant factors in the decision not to use this program.

2. There is always the option to *roll-your-own*. In this case an integrator and functions to evaluate a set of state-space equations are all that is needed. However, this is no small task. Writing all the support functions is a considerable effort. Experience has shown that 3-4 months could easily be spent writing all the code for a rigid-body analysis. The problem is there is then no room left for the errors or changes that will surely be made. However, if the time had been available this method may have been chosen. The resulting program, specifically tailored to this problem, may have been significantly faster than a program that utilizes a general formulation. This is a very import consideration if the program is going to be used on flight hardware. When flight hardware, and people are involved, safety is an important issue. There can be a tremendous amount of what-if games played; the result is the analyst gets to run hundreds, maybe thousands of simulations.

3. A commercially available tool such as ADAMS, SD-FAST, or DADS could be used. ADAMS has been used in the past for this type of analysis, and has proven quite useful. The fact that the analysis would be performed in two cities, and on several different computers was a severe disadvantage for the commercial codes. It would be a very expensive proposition to purchase 3-4 commercial licenses for 1 analysis task.

4. TREETOPS was considered for several reasons: TREETOPS uses a formulation based upon Kane's method, and therefore does not need to use additional equations for constraints between bodies such as pin-joints. See References [1] through [2]. One draw-back to this method is the assembly of a system mass matrix. As the size of the simulation grows the inversion of $[M]$[4] grows like N^3.

   TREETOPS has a menu-driven preprocessor which performs some error checking on the inputs. In many cases, the model is completely defined in the input file. The point being that an analyst can usually get a faster start if he/she does not have to write and debug code at first.

   TREETOPS is freely-available. Marshall Space Flight Center (MSFC) will supply interested parties with a tape of the FORTRAN source code. The recipient has the freedom to compile and use the software on any available computer. There are no restrictions to number of users or installations. The only restriction is that you do not re-distribute it; this is to help MSFC keep track of usage, and versions of the code. Additionally TREETOPS capabilities put it in the same class as the commercially available codes as far as capability (it is lacking in the graphical-user-interface dept.).

For this project TREETOPS was chosen. We knew we would probably have integration problems with almost any simulation tool we choose due to the small masses of the capture-latch mechanisms, and the high frequencies that contact forces usually introduce.

4 Computational Issues

There is not enough space here to discuss all aspects of the analysis, or even cover it in a chronological overview. Instead we will focus on those aspects

[4] An N-by-N matrix.

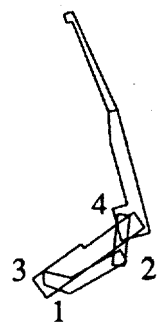


Figure 4: Capture Latch Mechanism, TREETOPS Model

that were either a source of difficulty, or required a large amount of the analyst's time.

4.1 Results Visualization

Interpretation of the simulation results, especially for a 3-dimensional analytical simulation, is often an overlooked issue. A great portion of time is occupied with interpreting simulation results. Not only does a good visualization tool speed up the process, but also reduces the likelihood of errors. To illustrate the point ... Figure 4 contains a simple representation of the TREETOPS mechanism model with the joints numbered. Figure 5 shows time history results of the joint relative euler angles. With this data alone it is difficult, and time consuming to verify that the output is correct. Especially since all the joint angles are not directly related to a either common frame, or a common body. Now add graphical animation output (see Figure 6) to the information at hand. As most will agree, the same conclusion can be arrived at quicker, and with more confidence if Figure 6 is available.

4.2 Numerical Integration

Throughout this analysis there were constant problems with the integration[5] step-size. Every time a new contact force model or a new simulation condition was introduced, the trial and error process of determining an adequate

[5]The standard TREETOPS integrator is a 4-th order Runge-Kutta

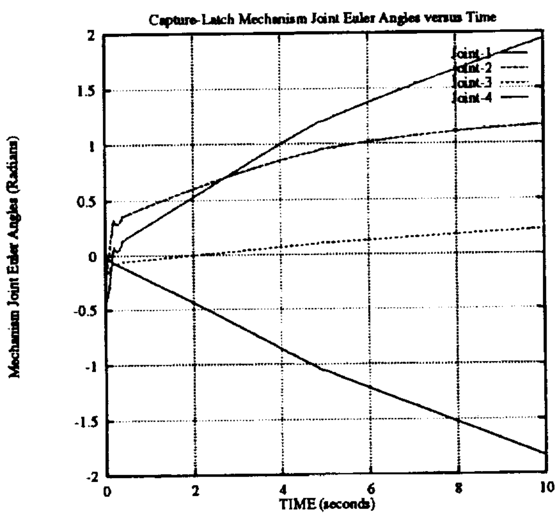


Figure 5: Capture-Latch Euler Angles

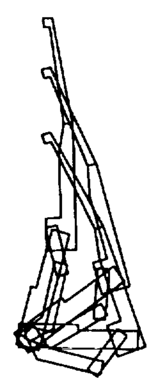


Figure 6: Capture-Latch Animation

step-size was repeated. An initial step-size of $\frac{1}{10}$ the period of the highest frequency component in the model was used repeatedly. However, this method often left us with a step-size that was too small. How can a step-size be too small? When the required simulation duration is 15-25 seconds (real time) a very small step-size can result in hours of cpu-time; this is unacceptable in the development phase of the analysis.

An additional factor interfered with our efforts to find a reasonable step-size. The difference between largest and smallest mass values was eight orders of magnitude. Since the mass matrix is inverted at every integration step, an ill-conditioned mass matrix can cause run-time problems that exhibit symptoms similar to step-size problems. When the simulation is unstable TREETOPS will often expire with the message "Mass matrix not positive definite". Since the mass matrix is configuration dependent, this implies that the system has reached a numerically impossible state.

Since there were two significant contributors to our integration problems, we decided to eliminate one of the sources. Since the mechanism velocities are small we decided to test the assumption that the mechanism component mass properties played a small role in the overall system dynamics by increasing the mechanisms mass properties by two orders of magnitude. Simulation comparisons with earlier simple mechanism models showed that this assumption was reasonable for some conditions. However, we must be careful and use this assumption only as a *stop-gap*, and not a permanent fix.

4.2.1 Contact Modeling

The contact modeling effort deserves some attention because at least 50% of the total effort was spent in the pursuit of accurate contact models. Not only did the contact models take a significant amount of man-power to produce; they also require a significant amount of computer time to simulate. For this problem there are 5 plausible types of contact:

1. Latch-arm / capture-hook.
2. Passive port ring / active port ring.
3. Passive port alignment guide / active port alignment guide.
4. Passive port alignment guide / active port ring.
5. Active port alignment guide / passive port ring.

For each type of contact, specific geometric calculations are performed to check for interference, or constraint violation on a case-by-case basis. If interference is detected a reaction force is computed. The reaction force is a function of the interference and the interference rate. We will look briefly at the alignment guide/guide contact model as an example.

The guide/guide contact is modeled with an edge/edge contact model. Figure 7 contains a picture of the model nodes, local coordinate systems and vectors used in the contact force computation. The procedure uses the cross product of two vectors, each representing an edge to facilitate computation of contact forces. The logic used follows:

1. Compute $\boldsymbol{C} = \boldsymbol{R}_a \times \boldsymbol{R}_b$ where $\boldsymbol{R}_a = \boldsymbol{A1} - \boldsymbol{A2}$ and $\boldsymbol{R}_b = \boldsymbol{B1} - \boldsymbol{B2}$. The shortest distance between the two lines will have the same direction as $\boldsymbol{C}$.

2. Transform the position and velocity vectors of the edge endpoints into the new coordinate system defined by the direction cosine matrix $\boldsymbol{A}$, formed as follows:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{C} = \boldsymbol{R}_a \times \boldsymbol{R}_b \\ \hline \boldsymbol{R}_a \\ \hline \boldsymbol{C} \times \boldsymbol{R}_a \end{bmatrix}$$

3. The shortest distance between $\boldsymbol{R}_a$ and $\boldsymbol{R}_b$ is the distance between $\boldsymbol{A1}$ and $\boldsymbol{B1}$ along the $\boldsymbol{C}$ axis. If that distance is less than zero:

   (a) Compute point of intersection:

$$\boldsymbol{intx} = -y_1 \left(\frac{x_2 - x_1}{y_2 - y_1} \right) + x_1$$

   (b) Calculate offsets:

$$\boldsymbol{aoff} = \boldsymbol{intx} \cdot \boldsymbol{R}_a$$

$$\boldsymbol{boff} = \left((\boldsymbol{intx} - B1(2))^2 + (B1(3))^2 \right)^{1/2} \cdot \boldsymbol{R}_b$$

   (c) Calculate relative velocity at point of intersection:

$$\boldsymbol{V}_a = \boldsymbol{V}_{a1} + \boldsymbol{\omega}_a \times \boldsymbol{aoff}$$

$$\boldsymbol{V}_b = \boldsymbol{V}_{b1} + \boldsymbol{\omega}_b \times \boldsymbol{boff}$$

   where $\boldsymbol{V}_{a1}$, $\boldsymbol{V}_{b1}$, $\boldsymbol{\omega}_a$, and $\boldsymbol{\omega}_b$ are the velocities at points A_1 and B_1. These velocities are supplied by TREETOPS.

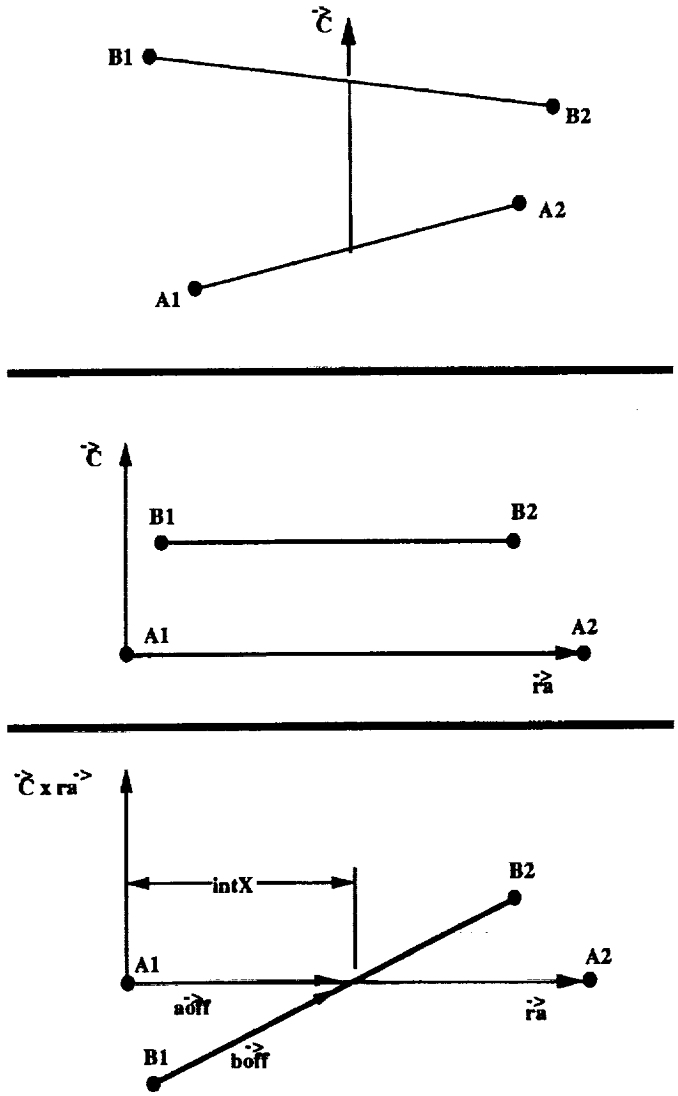


Figure 7: Guide - Guide Contact Geometry

The above procedure calculates the relative distance and velocity between the two edges. A FORTRAN subroutine that performs this task can run anywhere from two to four pages (including comments). Once the contact geometry is defined the remaining task is to define the physical properties of the contact.

For elastic contact at low velocities it may be acceptable to use a simple linear[6] spring and damper to model the impact effect. Calculation of the proper stiffness can usually be accomplished via careful static analysis of the involved components. One can take the local stiffness, and combine it in series with the global stiffness of the contacting parts. In this case the load path we are concerned with is the in-plane loading of the edges of an alignment guide. Like the integration step size, calculated contact stiffness values are only a starting point. Once the initial value is calculated it must be tested. Insight into the appropriateness of the initial value can be gained by observing the measured deflections at the point of contact.

4.3 Back-Driving the RMS

Modeling and simulating the RMS back-drive forces was one of the simpler aspects of the analysis. Why is such a complex piece of hardware simple? Neither the SSRMS or the SRMS was modeled in any detail. Instead a *requirement-model* was assembled. A requirement-model does not necessarily behave like the physical component it is supposed to represent; instead its behavior is representative of the worst-case events or conditions as spelled out in the requirements document. In this case the requirement-model of the RMS was an arm that supplied a constant 150 lbf resistance in all directions.

Simulations showed that the existing capture-latch mechanisms could not successfully close the gap between the active and passive ports. Indeed simulations showed that a mechanism with 3 times the authority was required to pull the two ports together under extreme misalignment conditions and worst case contact friction and RMS back-drive forces.

5 Summary

We have discussed various aspects and problems associated with a 3-4 month analysis effort. Unfortunately there is not room enough to discuss all of the problems involved. From a computational point of interest it is important to note:

[6] linear in the sense that the spring has a constant stiffness when engaged

1. Data visualization tools (VDS, BPLOT) are crucial to performing a timely analysis. As simulations get more and more complex, so must the output visualization capability.

2. The method used herein for calculating contact forces is much too tedious, time consuming, and error prone. If this method proves to be the most computationally efficient, then some way of speeding up the algorithm development and debugging is in order.

3. It is important to select the software tool best suited for a particular job. Therefore, a collection of specialized tools like TREETOPS, that solve a certain class of problems well, is essential. Our experience shows that software that tries to do everthing, ends up doing nothing well.

6 Acknowledgements

Thanks to Bill Gustafson for always providing interesting work. We would like to thank Phil Williams and Pat Tobby from Control Dynamics Company, who have, and continue to develop a more detailed simulation capability for Boeing, and MSFC at the Marshall 6-dof facility. They provided invaluable assistance during the analysis.

References

[1] *User's Manual for TREETOPS, Revision 8*, Singh Likins, et al., NASA contract no. NAS8-34588.

[2] *Dynamics of Flexible Bodies in Tree Topology*, R. P. Singh and R. J. VanderVoort, Journal of Guidance, Control, and Dynamics, Sept-Oct 1985.